Robotic Collision Avoidance Simulation Using a Two-Wheeled Robot and LiDAR Sensor

Mary Stirling Brown

Samford University  
800 Lakeshore Dr  
Birmingham, AL 35229  
+1 251 458 5466

mbrown35@samford.edu

Brian Toone

Samford University  
800 Lakeshore Dr  
Birmingham, AL 35229  
+1 205 726 2960

brtoone@samford.edu

**ABSTRACT**

Autonomous vehicles typically use a variety of sensors, including LiDAR sensors, to navigate through an environment and avoid obstacles without intervention. A LiDAR works by measuring the time it takes for the radiated laser to be reflected off an object back to the sensor in order to obtain a distance measurement to objects in the environment. In this research, a two-wheeled robot with a single-layer, rotating LiDAR sensor is used to test a collision avoidance algorithm in a simulated environment using Webots software. The collision avoidance algorithm is based on algorithms used in previous research. The created algorithm analyzes the LiDAR data to turn a specific angle away from the object to reach its target destination without collision. The robot has GPS and compass to aid in navigation towards its target destination. An environment with various obstacles is set up to test the algorithms. The proposed algorithm was unable to be tested due to a software bug and time limitations.

**Categories and Subject Descriptors**

**[Computer Systems Organization]:** Embedded and Cyber-physical systems – *robotics, robotic autonomy, sensors and actuators.*

**[Theory of computation]**: Data structures and algorithms for data management.

**General Terms**

Algorithms, Design, Experimentation

# INTRODUCTION

Autonomous robotics has spread to many different sectors of people’s everyday lives, from autonomous cars, industrial manufacturing, food deliveries, and land mapping and surveying. Collision avoidance algorithms is a current research topic in Computer Science to enable robots to autonomously navigate. Various sensors, such as sonar, light sensors, cameras, and many more, have been used in this research. Another viable option is a LiDAR sensor, standing for Light Detection and Ranging. The LiDAR shoots pulses of laser light to find distance and other information from the target. It calculates the distance from an object by measuring the time a laser is detected from the sensor to being reflected to the sensor [2].

The next section of this paper details the background with a literature review. Section 3 focuses on the methodology and algorithms that are implemented in this research for a successful collision avoidance system. This procedure is tested in a simulated environment, and its results are presented in Section 4. Finally, Section 5 talks about the conclusions of this research as well as limitations and future work.

# BACKGROUND

According to Hutabarat, *et. al*, the LiDAR technology proved to be successful in that the robot was able to avoid objects of certain colors in its navigation. It does this by using the Braitenburg algorithm where sensor input affects the motor speed. This research concluded that the color of an object and the intensity of ambient light do not affect the distance measurements taken by the LiDAR [2].

In research by Baras, *et. al*, a robot is successfully able to navigate in an unknown environment by only using a single LiDAR in their robotic design with no other sensors. The robot in their tests was able to avoid obstacles through the implementation of a point cloud from the LiDAR data [3].

According to Wu, *et. al*, a collision avoidance algorithm is developed based off a median filter that replaces a group of points with a midpoint. The scanned area is then grouped by emergency, accurate, and fuzzy obstacle-avoidance areas. Tests revealed that this approach was more efficient than other single collision avoidance algorithms [4].

Research by Park, *et. al*, reveals that the Artificial Potential Fields algorithm works by the robot being attracted to its waypoint, or target destination, and repelled by a negative charge, or obstacle. There are limitations to this when the robot gets trap in local minima of an environment [5].

# METHODOLOGY AND ALGORITHMS

## Hardware and Software

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*Samford University Department of Math and Computer Science*, May 2011, Birmingham, AL. Copyright 2011 Samford University…$10.00.

robot simulation software. This software allows for the development of different 3-D environments with different physical factors, such as mass, gravity, friction, and many more. All this information is stored in a *world*. Robots in this world can be programmed by creating a *controller* with various programming languages. In the case of this research, Python is used to code the robot [1].

The robot used in this research is Parallax’s Boe-Bot. It is a three-wheeled robot with two motorized wheel and one passive wheel. It includes many sensors, including distance sensors, position sensors, touch sensors, and light sensors.

The LiDAR designed in this research is fixed on the front side of the robot. It was built with the features of having a single layer of scanned data with a max distance range of five meters. It has a 90-degree field of view with 25 distance readings for one layer at a time.

## Collecting the LiDAR Distance Data

The LiDAR data scale ranges 90° with 25 distance readings per scan. The angle between each distance measurement was calculated by:

(𝑓𝑖𝑒𝑙𝑑 𝑜𝑓 𝑣𝑖𝑒𝑤)/(ℎ𝑜𝑟𝑖𝑧𝑜𝑛𝑡𝑎𝑙 𝑟𝑒𝑠𝑜𝑙𝑢𝑡𝑖𝑜𝑛)=(90°)/25≈3.59817°

A counter variable keeps track of which distance reading was happening at each data point, from 1 to 25. The overall angle of that distance measurement is added up to get the measurement from the 0° angle, which is the first distance measurement in a complete scan.

The LiDAR measures the distance of an object in meters. This information and the angle measurement is stored in a Python tuple as (angle measurement from 0°, distance from objects in meters).

## Differential Drive Method

Include additional sections here for describing your algorithm or software.

Table . Here is an example table. The caption should be centered above the table like this. The content of the table is just a few selected projects with topic area and platform used.

|  |  |  |  |
| --- | --- | --- | --- |
| **Brown** | **Barnett** | **Wagoner** | **Hixson** |
| Distributed Computing | Mobile | Mobile | Social Web2.0 |
| Clusters | Appcelerator | Marmalade | CakePHP |

## Collision Avoidance Algorithm

This section will detail the collision avoidance algorithm that I create in my project. I plan to use the Artificial Potential Field algorithm and a few other algorithms to create it. I plan to mimic many of the designs of the Artificial Potential Field algorithm. To do this, I will implement similar code where the robot is negative repelled from obstacles and positively attracted to its target destination. I want to create my own algorithm in this way. This will be the bulk of my project for testing and implementing it to make it the most efficient and successful.

# Testing and Results

Describe how you tested your software. Describe the intended audience and the delivery that you made of your software to that audience. If you haven’t made this delivery yet, then describe how you will deliver your software.

# CONCLUSION

Describe your conclusions in this section.

## What you learned

Include this section describing what you learned while completing this project. Normally, you

## Where to go from here

Describe what interesting research areas your work has opened up. Describe what you would do if you had more time to work on this project.

# ACKNOWLEDGMENTS

This research is supported by the Department of Mathematics and Computer Science at Samford University.

# REFERENCES

1. Albert Alfrianta. 2021. Simple avoidance algorithm, implemented on e-puck robot and simulated on Webots Robot simulator. (August 2021). Retrieved April 14, 2022 from https://medium.com/@albert.brucelee/3143c096d285
2. Cyberbotics. 1998. Webots: robot simulator. Retrieved from https://cyberbotics.com#cyberbotics
3. D. Hutabarat, M. Rivai, D. Purwanto and H. Hutomo, "Lidar-based Obstacle Avoidance for the Autonomous Mobile Robot," 2019 12th International Conference on Information & Communication Technology and System (ICTS), 2019, pp. 197-202, doi: 10.1109/ICTS.2019.8850952.
4. Gada, Kajal. *How to Use Lidars in Webots? // Webots Lidar Code in Python Tutorial*. *YouTube*, YouTube, 9 Sept. 2021, https://www.youtube.com/watch?v=7A9h7d-XWSA. Accessed 14 Apr. 2022.
5. Min Gyu Park, Jae Hyun Jeon and Min Cheol Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No.01TH8570), 2001, pp. 1530-1535 vol.3, doi: 10.1109/ISIE.2001.931933.
6. N. Baras, G. Nantzios, D. Ziouzios and M. Dasygenis, "Autonomous Obstacle Avoidance Vehicle Using LIDAR and an Embedded System," 2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST), 2019, pp. 1-4, doi: 10.1109/MOCAST.2019.8742065.Fröhlich, B. and Plate, J. The cubic mouse: a new device for three-dimensional iput. In *Proceedings of the SIGCHI conference on Human factors in computing systems   
   (CHI ’00)* (The Hague, The Netherlands, April 1-6, 2000). ACM Press, New York, NY, 2000, 526-531.
7. P. Wu, S. Xie, H. Liu, J. Luo and Q. Li, "A novel algorithm of autonomous obstacle-avoidance for mobile robot based on LIDAR data," 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2015, pp. 2377-2382, doi: 10.1109/ROBIO.2015.7419694.Sannella, M. J. *Constraint Satisfaction and Debugging for Interactive User Interfaces.* Ph.D. Thesis, University of Washington, Seattle, WA, 1994.